MACHINE LEARNING ASSIGNMENT 1

GANESHREDDY ANNAPAREDDY

Person Number : 50442295   —   **ganeshre@buffalo.edu**

SIMPLE LINEAR REGRESSION

In line with the observational data, regression models describe the relationship between the variables. Although logistic and nonlinear regression models employ a curved line, linear regression methods just don't. You may estimate a dependent variable's change as an independent variable or set of independent variables changes using regression. To determine the association between two quantitative variables, Simple Linear Regression is performed.

Formula :

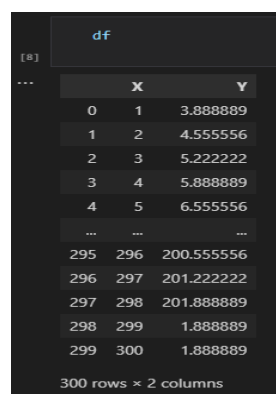$$y = \beta_0 + \beta_1 X + \varepsilon$$

Here :

- y is the dependent variable's (y) expected value for any given value of the independent variable (x)

- The intercept, or expected value of y when x is 0, is represented by β_0.

- The regression coefficient, or β_1, indicates how much we anticipate y to change when x rises.

- X is an independent variable.

- ε stands for the estimate error, or the degree of variance in our estimation of the regression coefficient.

Coding :

I have taken the Linear regression data set from the Kaggle an open source platform . The link https://www.kaggle.com/datasets/andonians/random-linear-regression . This is a CSV file containing 700 data pairs makes up the training dataset (x,y). The x-values are integers in the range of 0 to 100. The Excel function NORMINV(RAND(), x, 3), has been used to get the equivalent y-values. Therefore, x should be the best estimate y. A CSV file containing 300 data pairs makes up the test dataset.

STEP 1: Loading the data as df.

*Here we can see the x, y columns with 300 rows

STEP 2: I am training and testing the x & y values .

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size = 1/4, random_state = 0)
```

STEP 3: I am importing the linear regression from "sklearn.linear_model" and fitting the model to training set.

```python
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor = regressor.fit(X_train, Y_train)
```
[20]

After that I am going to predict the results.

```python
Y_pred = regressor.predict(X_test)
```
[21]

STEP 4: I am going to visualize the data using the scatter plot for both (x_train, y_train) and (x_test, y_test)

```python
plt.scatter(X_train , Y_train, color = 'red')
plt.plot(X_train , regressor.predict(X_train), color ='blue')
```
[22]

... [<matplotlib.lines.Line2D at 0x1ce9e3eb0a0>]


```python
plt.scatter(X_test , Y_test, color = 'red')
plt.plot(X_test , regressor.predict(X_test), color ='blue')
```
[23]

... [<matplotlib.lines.Line2D at 0x1cea04d4310>]